**Data Trained**

**Batch 1839**

**J USHARANI**

**Customer churn Prediction**

**Problem definition:-**

Customer churn is when a company’s customers stop doing business with that company. Businesses are very keen on measuring churn because keeping an existing customer is far less expensive than acquiring a new customer. New business involves working leads through a sales funnel, using marketing and sales budgets to gain additional customers. Existing customers will often have a higher volume of service consumption and can generate additional customer referrals.

Customer retention can be achieved with good customer service and products. But the most effective way for a company to prevent attrition of customers is to truly know them. The vast volumes of data collected about customers can be used to build churn prediction models. Knowing who is most likely to defect means that a company can prioritise focused marketing efforts on that subset of their customer base.

Preventing customer churn is critically important to the telecommunications sector, as the barriers to entry for switching services are so low.

This article is containing the following sub-topics

1. Problem Definition
2. How attrition impact the business
3. How to HR analysis help in understanding probable attrition case
4. Data Analysis

A. Understanding the data

3. EDA Concluding Remark

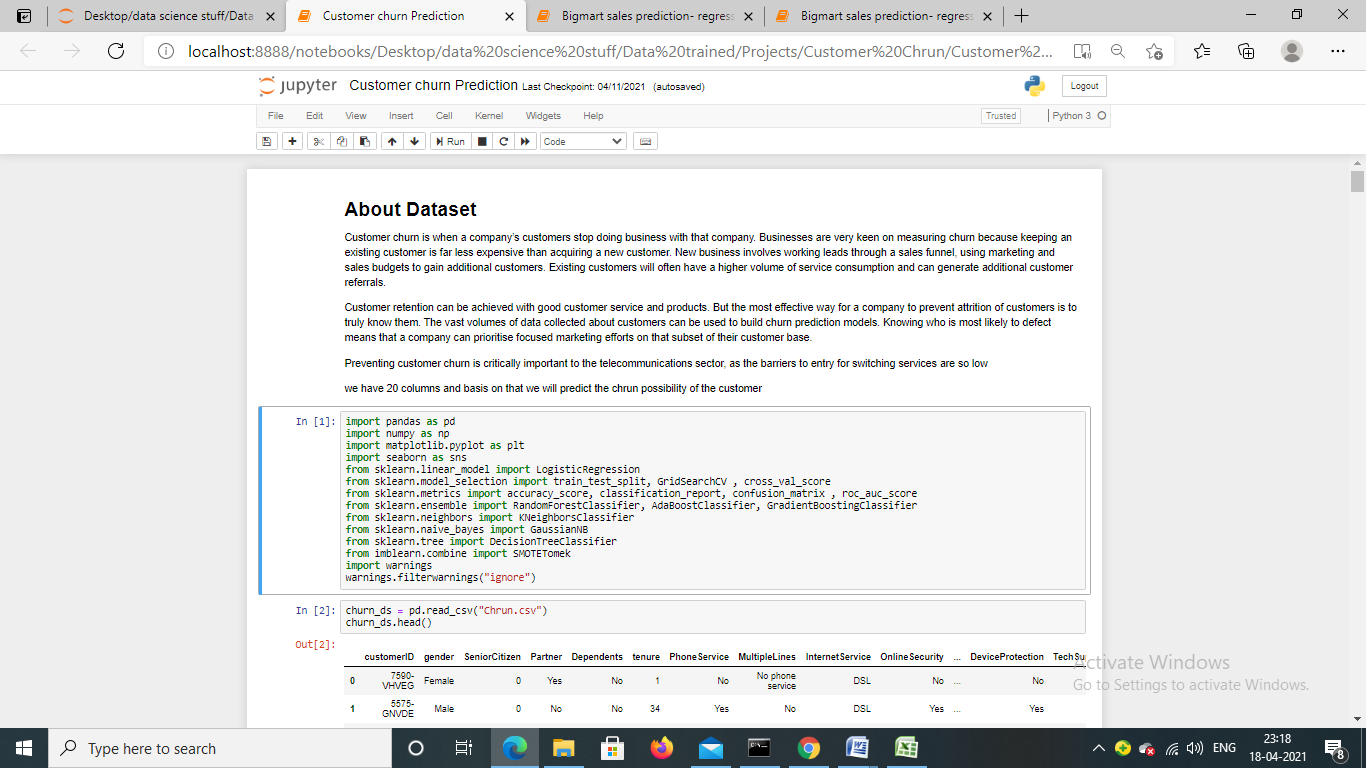
4. Pre-Processing Pipeline

5. Building Machine Learning Models

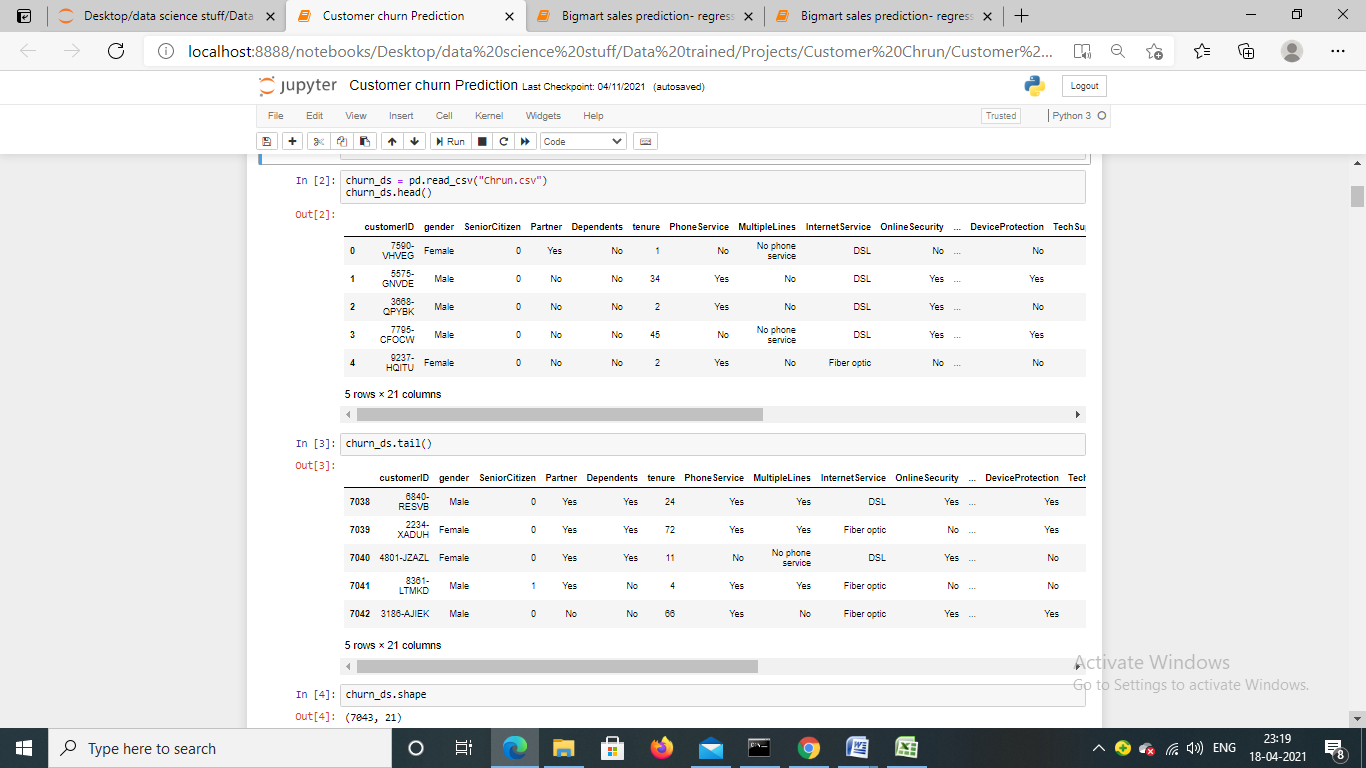
6. Concluding Remarks.

**Importing libraries:-**

We need some libraries to be imported to work upon on dataset, we would import dataset by using panda’s read\_csv method, if you want to refresher your idea about pandas, please visit pandas official site and documents.

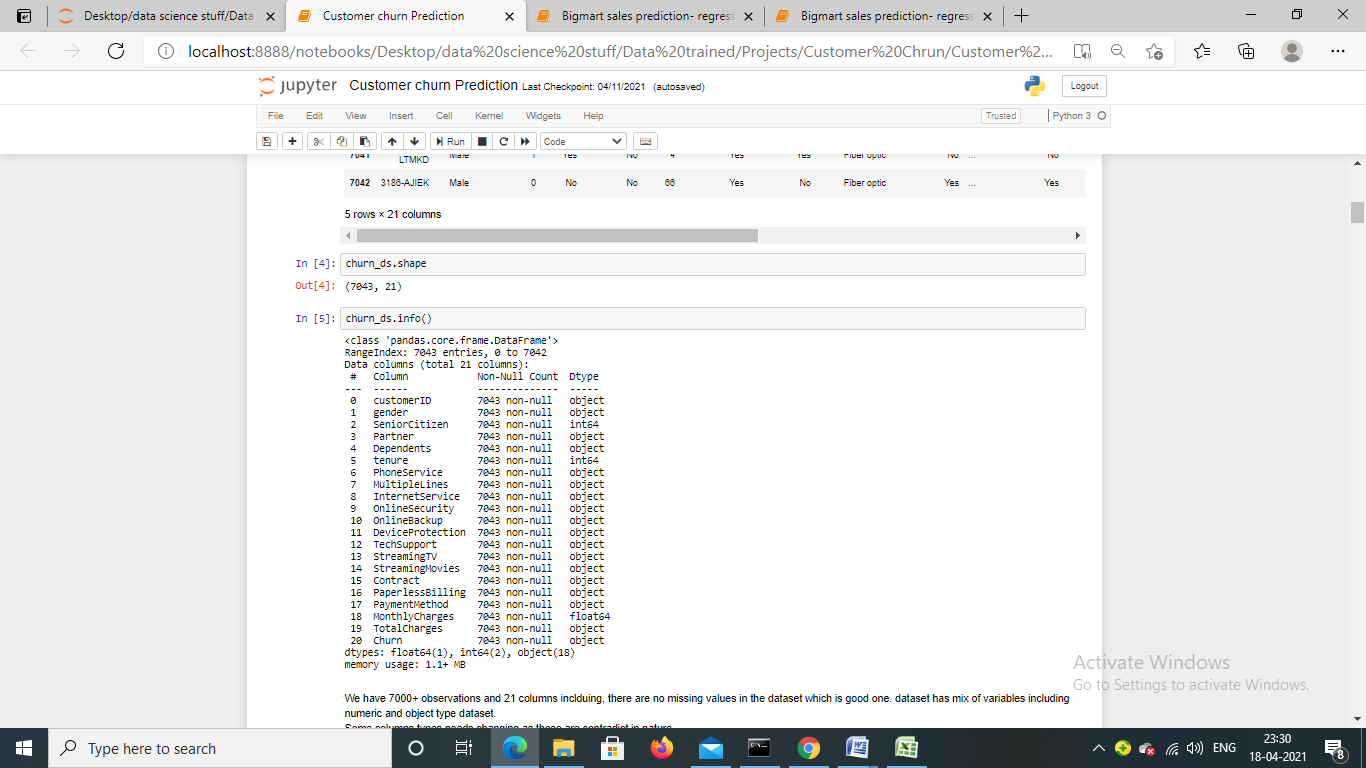
****

**Loading/gathering data:-**

****

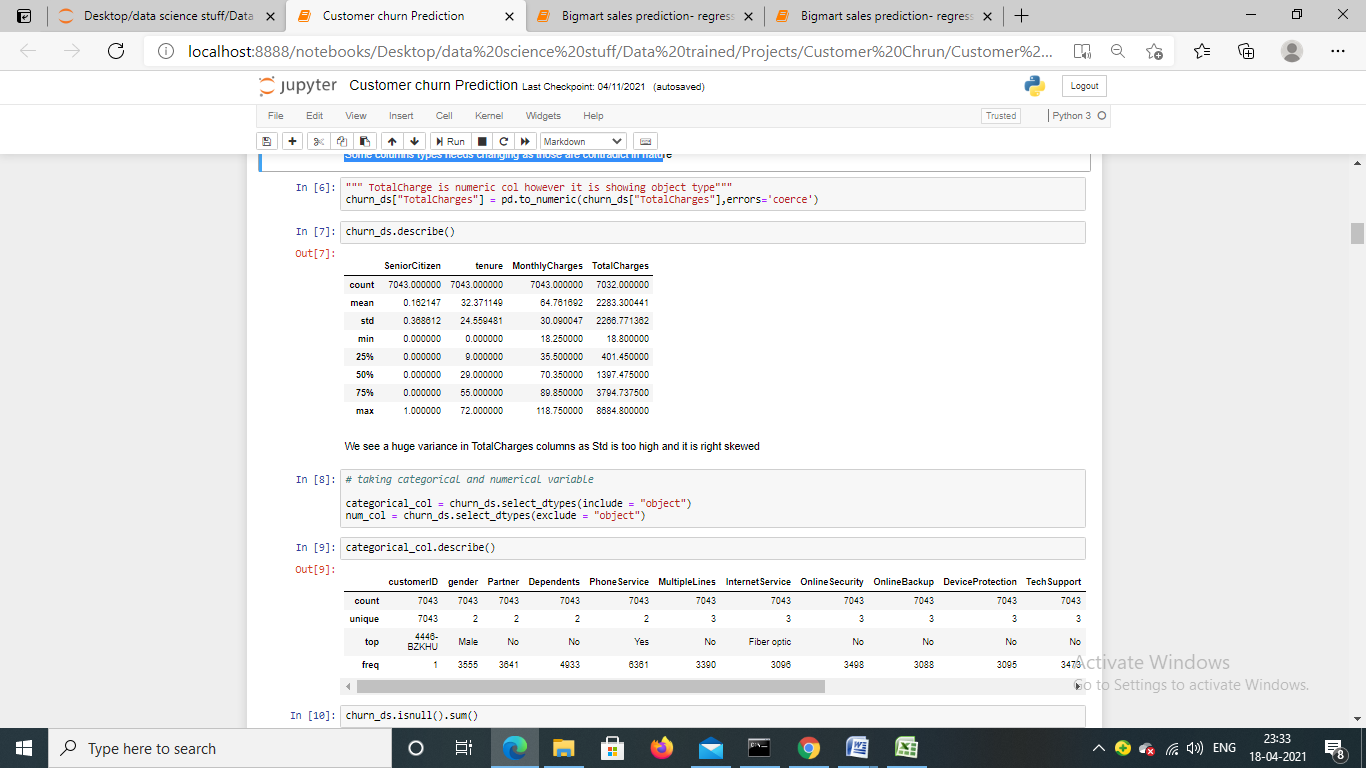
Dataset has been imported by using pandas read\_csv() function. We can see, it has mix of data types. Let’s check the shape of the dataset by calling shape method:-

* **Data Exploration/Analysis:-**

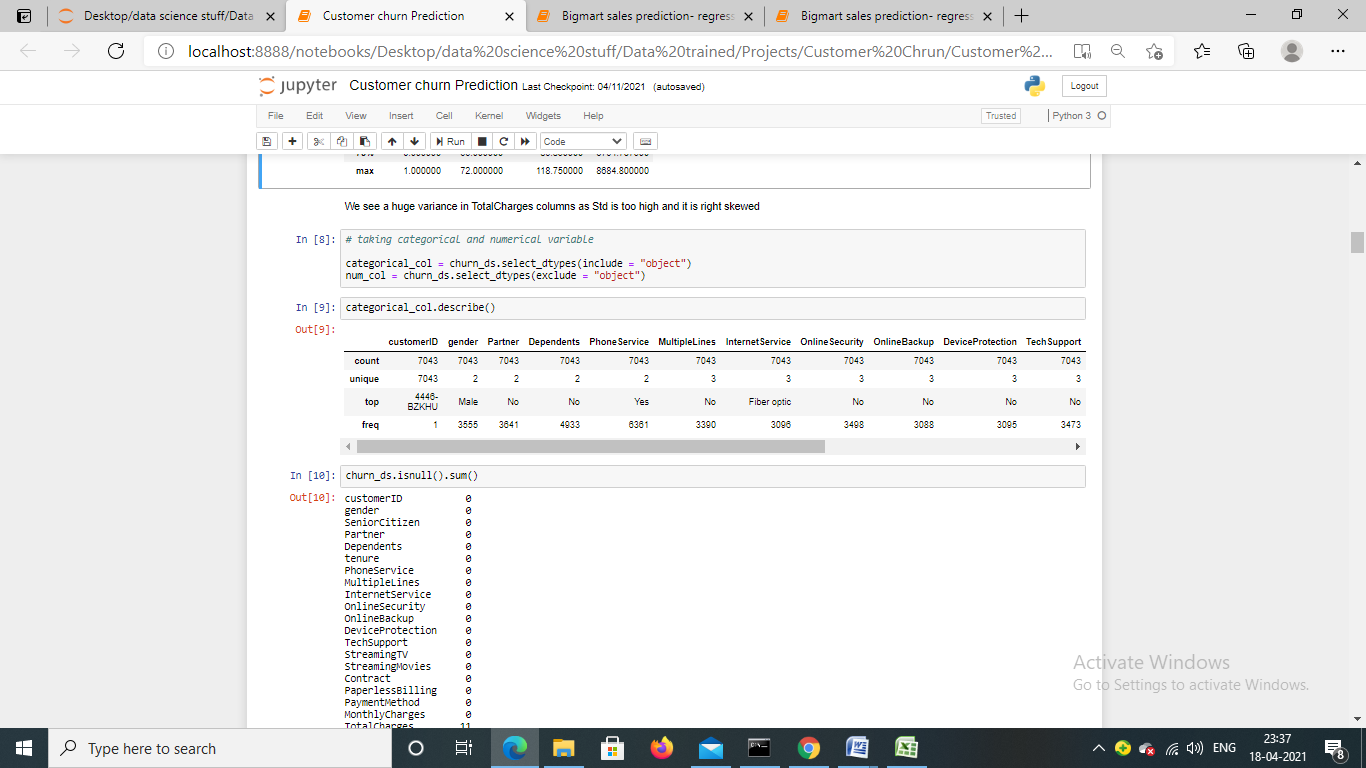


We have 7000+ observations and 21 columns including target variable, there are no missing values in the dataset which is good sign. Dataset has mix of variables numeric and object type. Some columns type’s needs changing as those are contradict in nature like total charges should be float rather than object.

**Using describe method to check statistical information of the numerical features.**

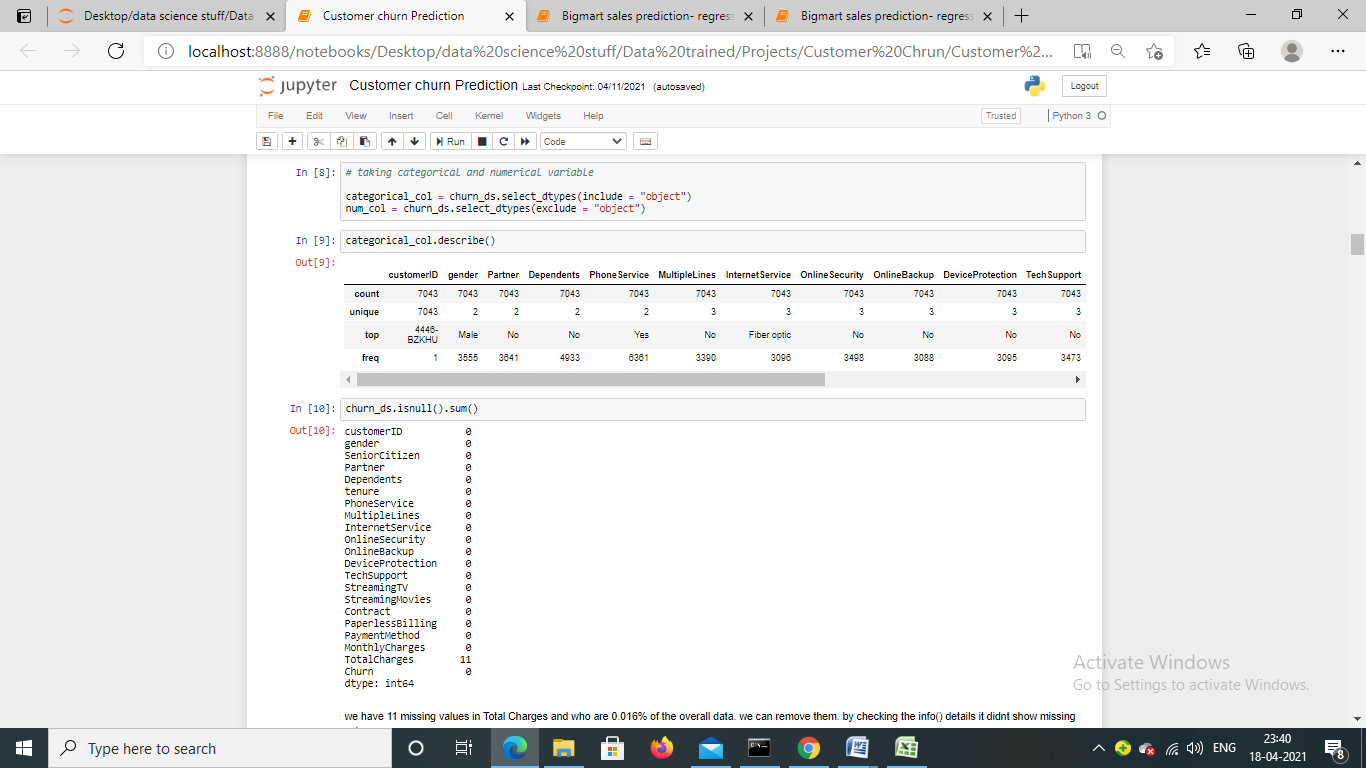


We see a huge variance in Total Charges columns as std() is too high and it is right skewed. We can see that there are people who have been with the company from long time as maximum tenure is 72.

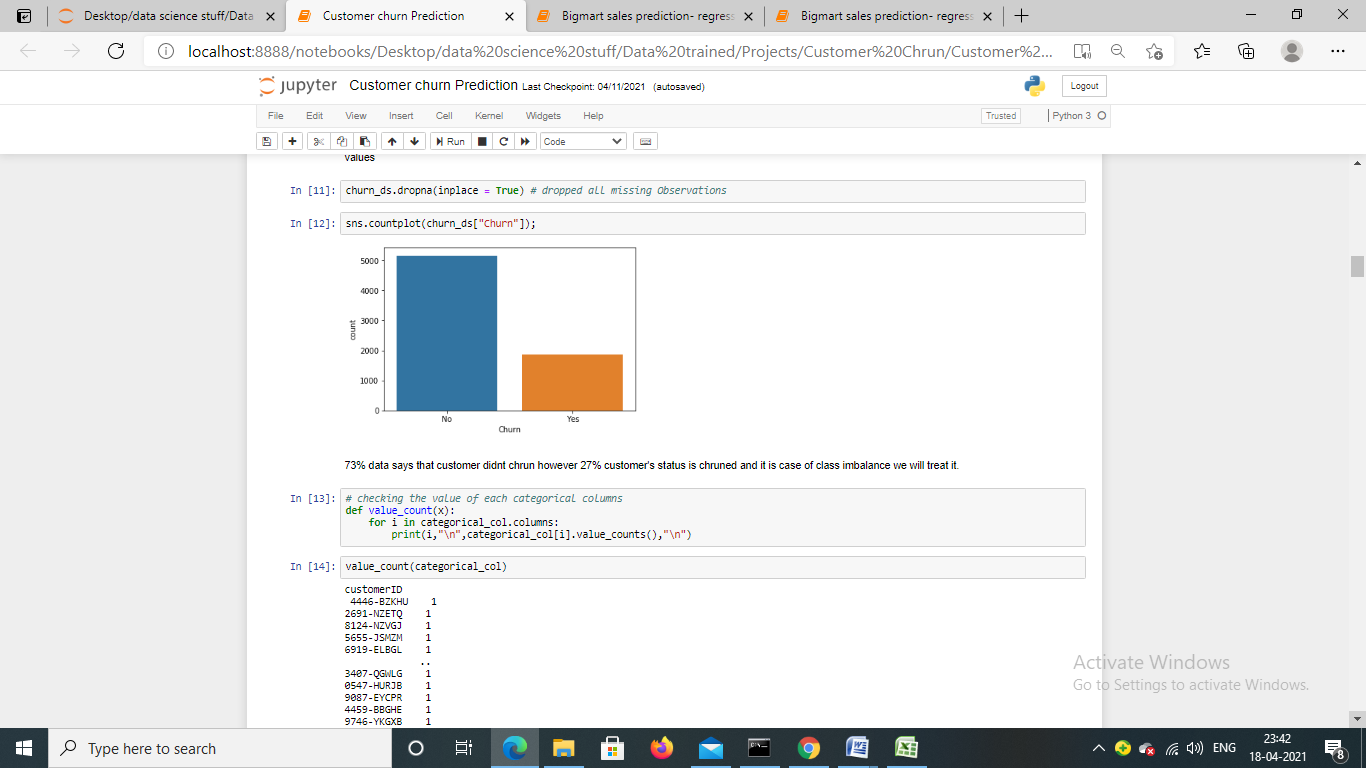


We are separating categorical and number column separately for further analysis. I have used describe method on categorical features as well and found that we have maximum 4 category in a column.

**Checking missing value**



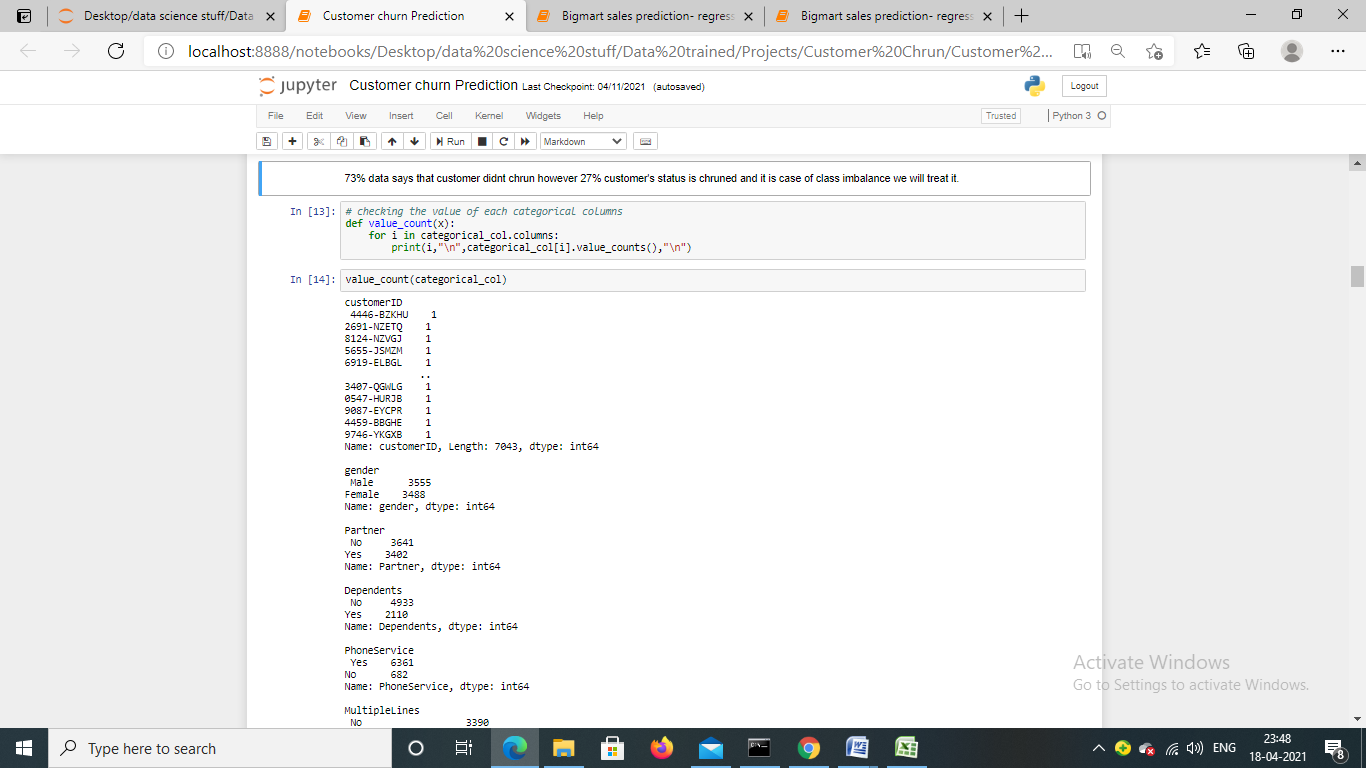
While checking the missing value, it has been found that Total charges column has 11 missing value which is 0.016% to the overall values. We will drop these values, by dropping these values we will not lose any information.



As we mentioned that we will drop missing value from the data. We are dropping them by using pandas dropna method, code snippet is attached.

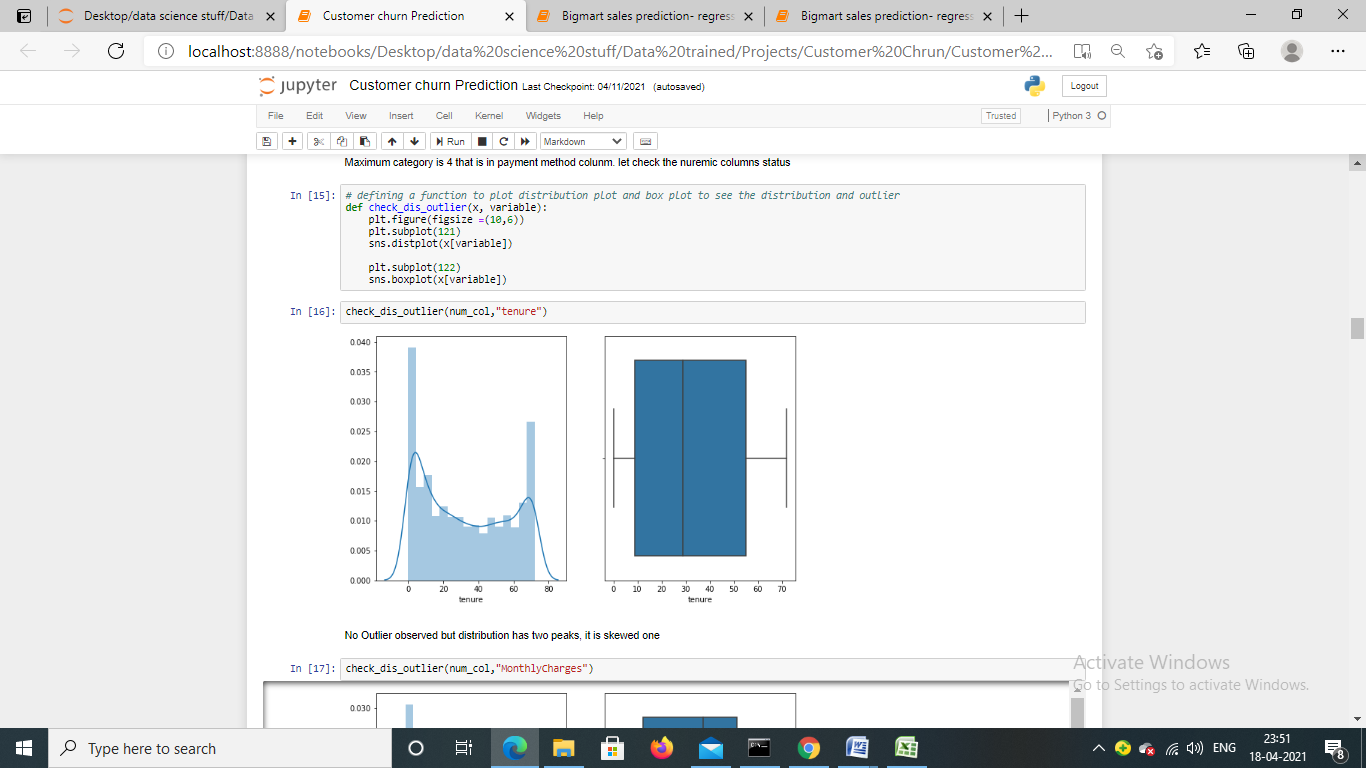
Post that i am checking class label proposition, looking at the proposition, it looks like class imbalance problem. 73% data says that customer didn’t churn however 27% customer's status is churned

**Let’s check categorical features unique values:-**



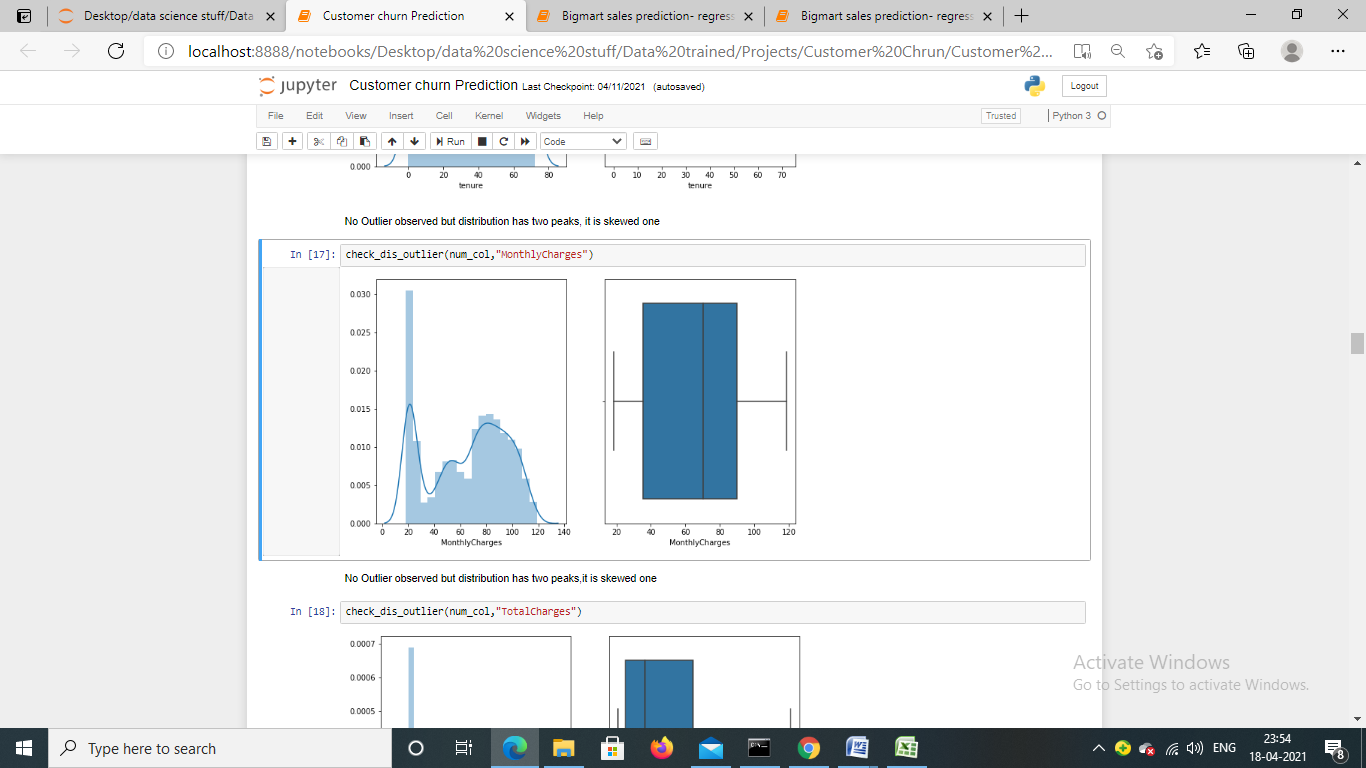
We have 18 categorical columns; we are showing a few of them. As we already mentioned that maximum category is 4.

**Let’s check the outlier:-**

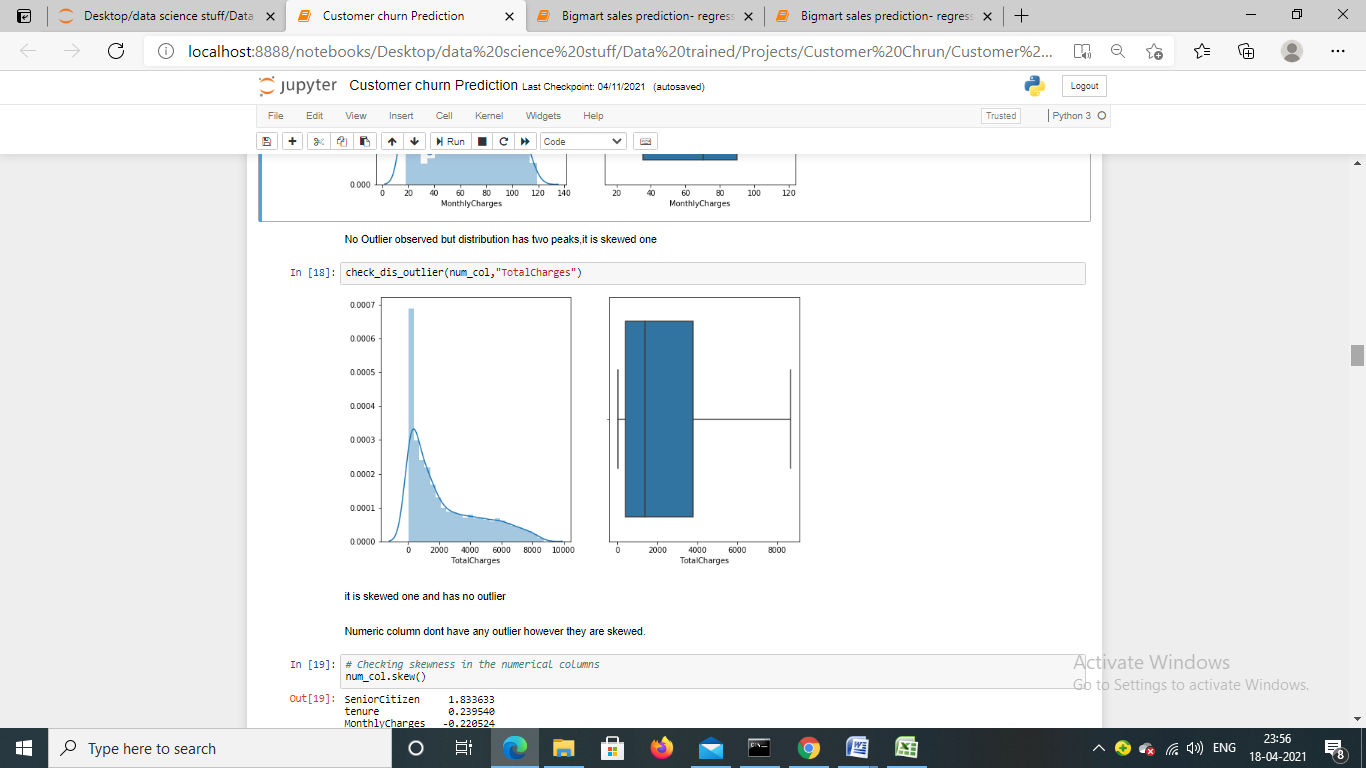
****

We are checking the distribution and outlier of numerical columns one by one, I am using seaborn distplot and boxplot method to visualize the distribution.

We can see that there is no outlier in tenure variable. Let’s check other features too.

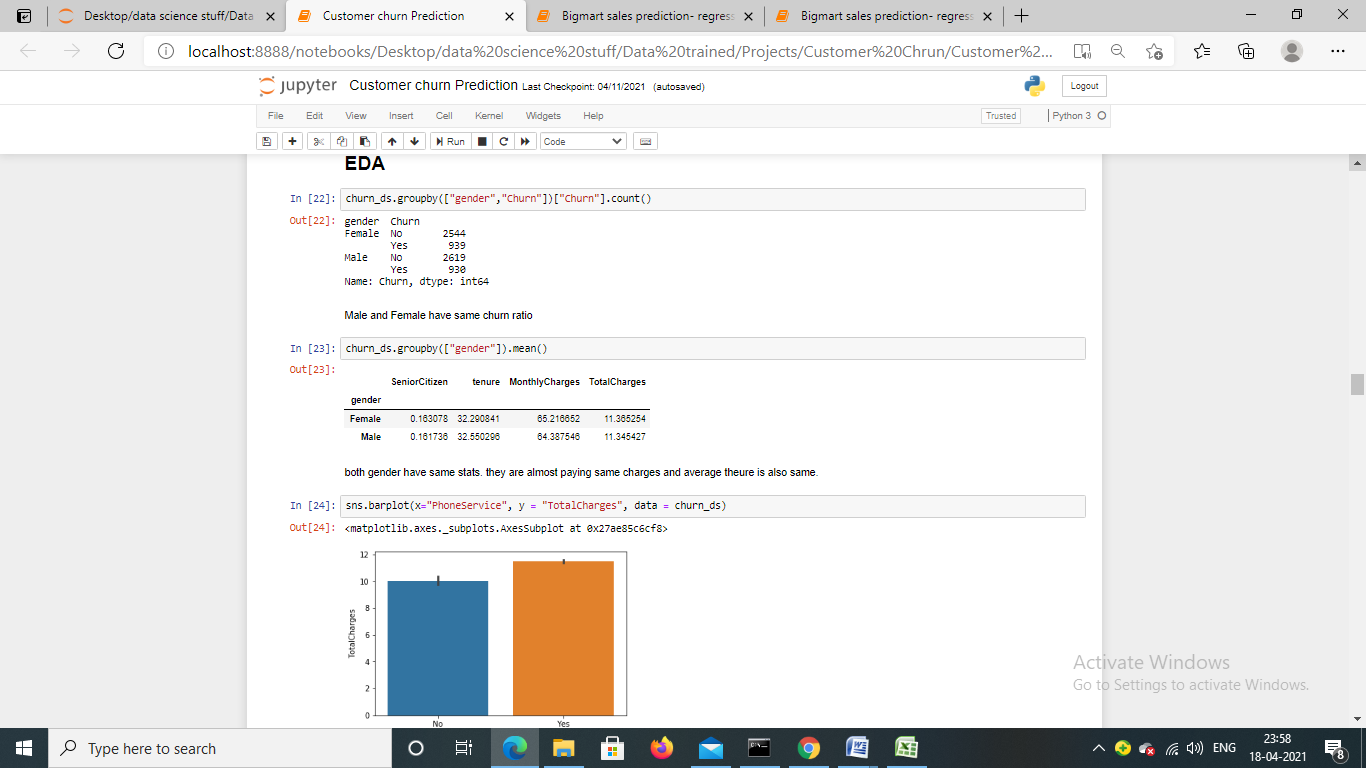


There is no outlier in monthly charges column and it is right skewed

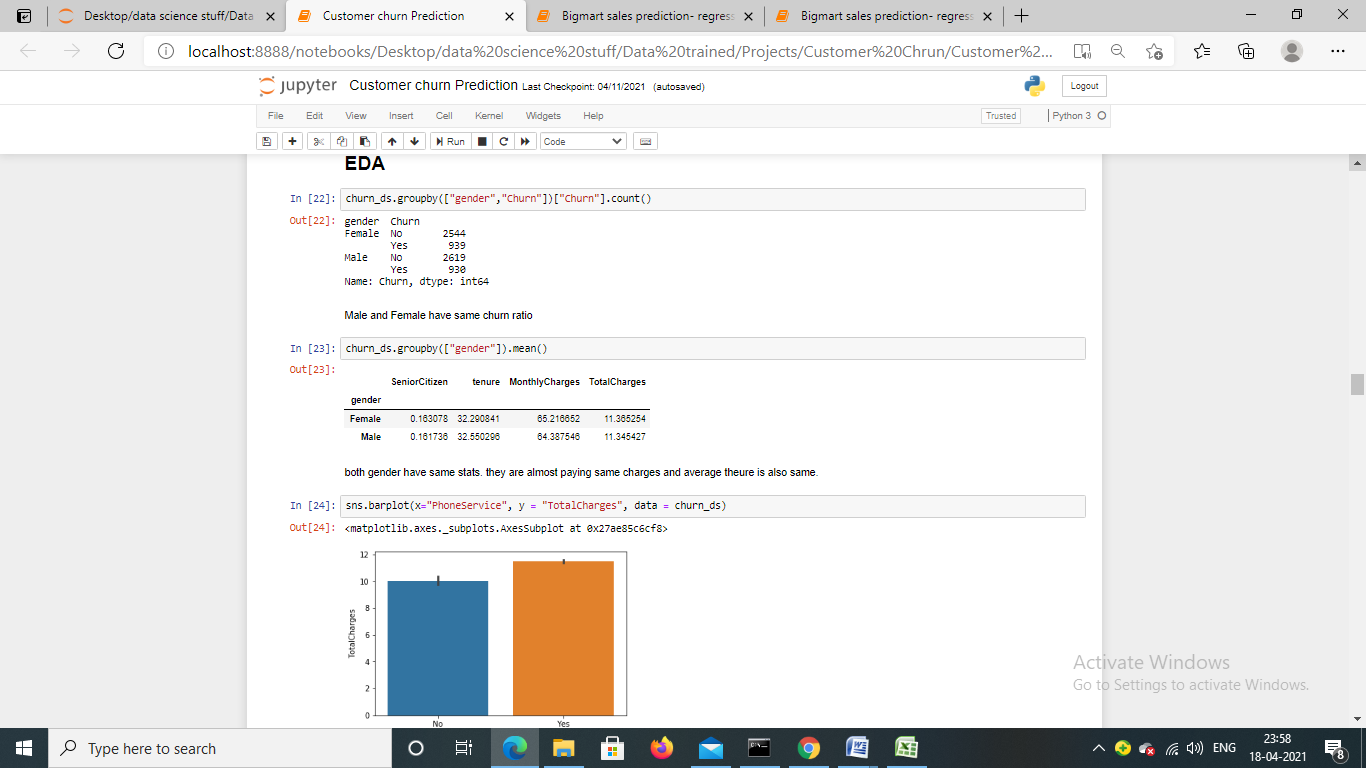


There is no outlier in Total charges column and it is also right skewed.

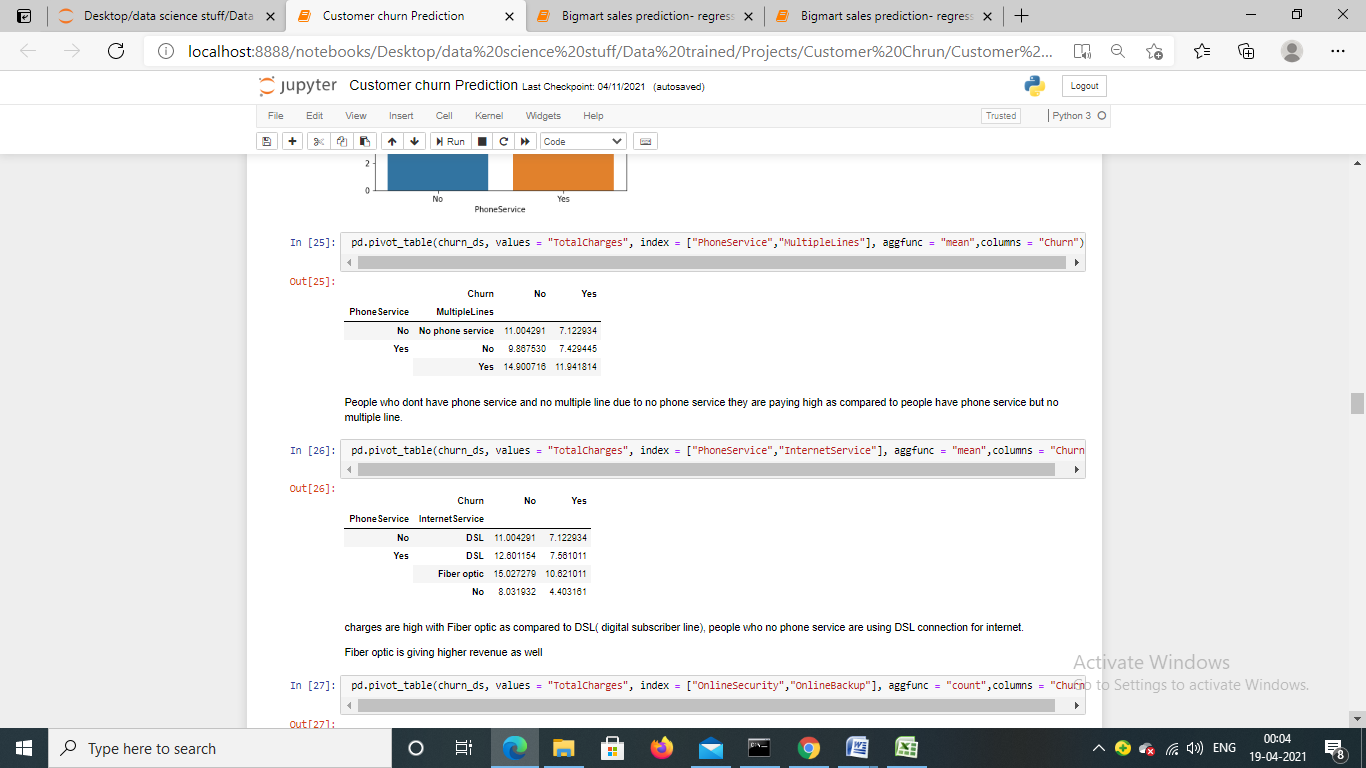
**Analysing churn:-**



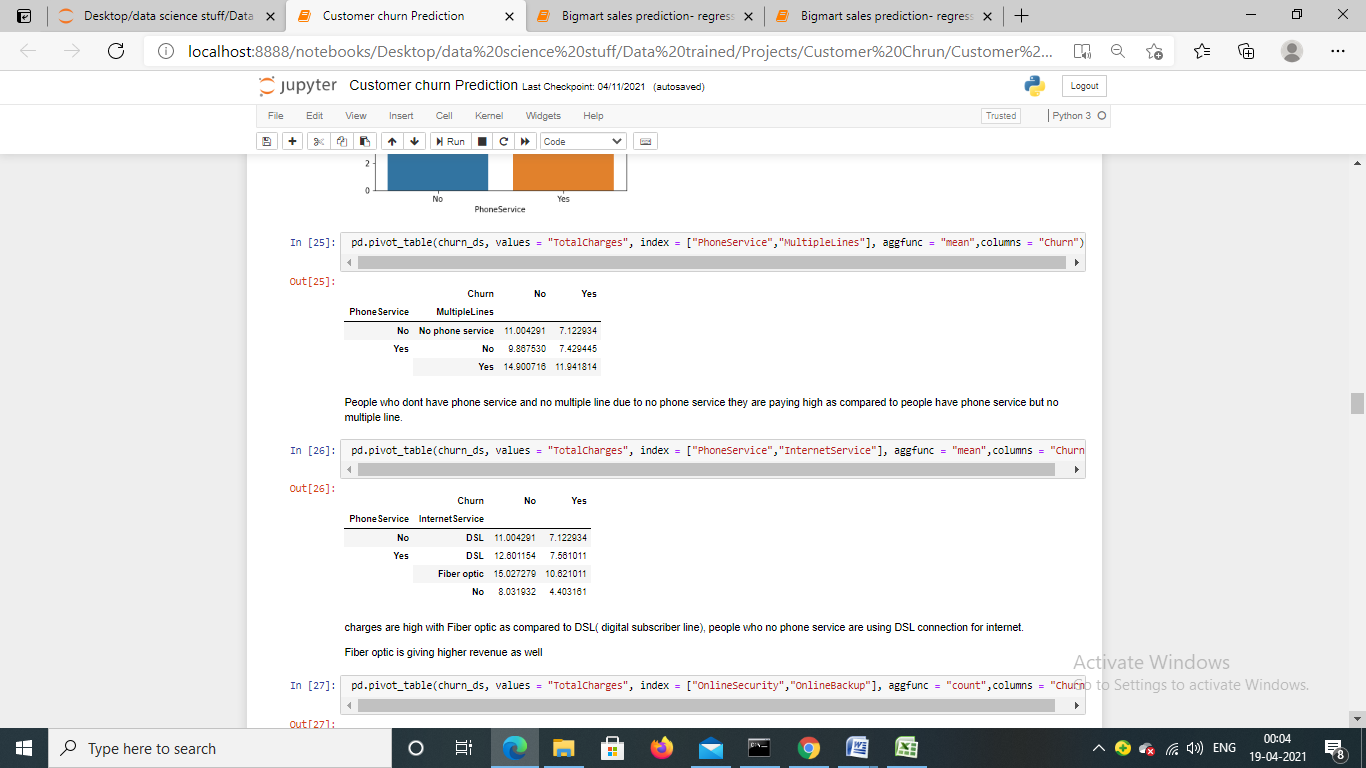
There is no significance difference in Male and female churn ratio.



Gender wise average total payment is also almost same. Which means, gender has no relation with churn

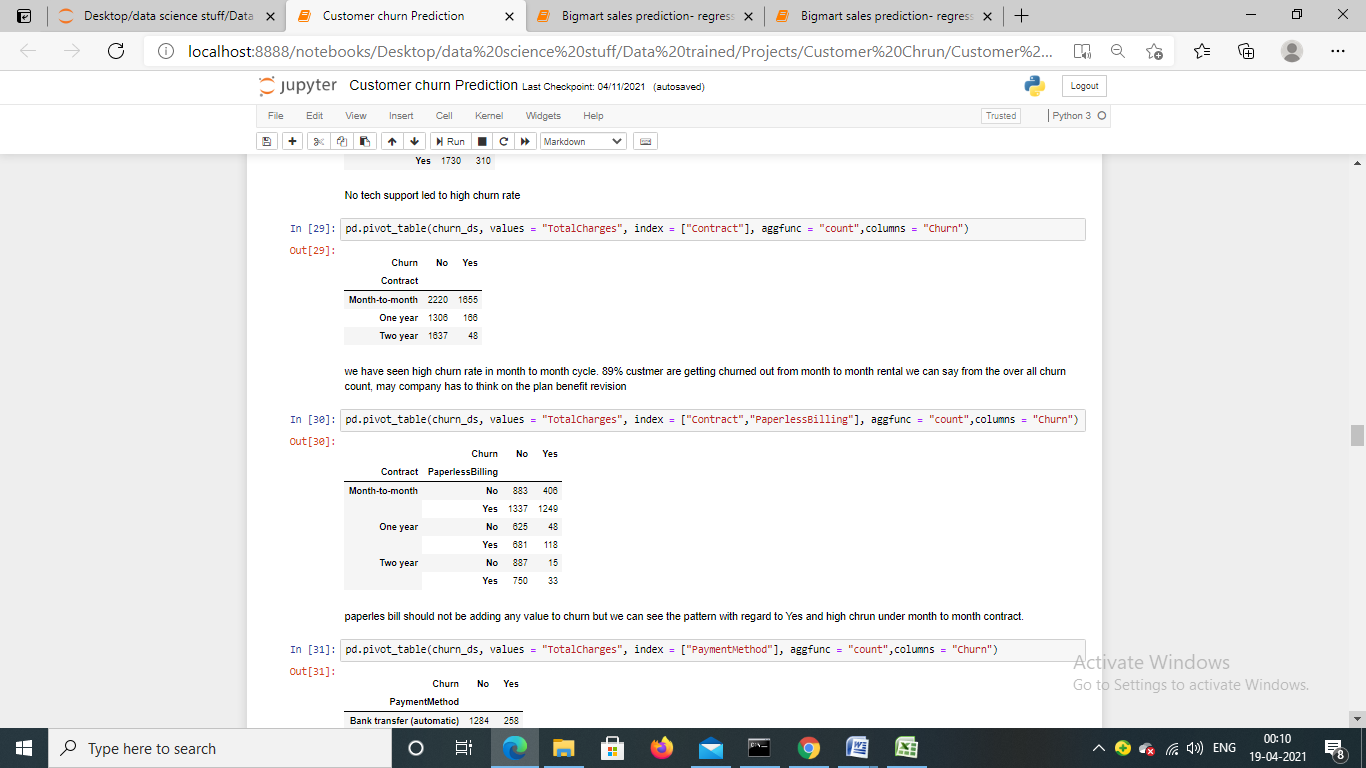


People who don’t have phone services and no multiple lines due to “no phone service” they are paying high as compared to people have phone service but no multiple lines



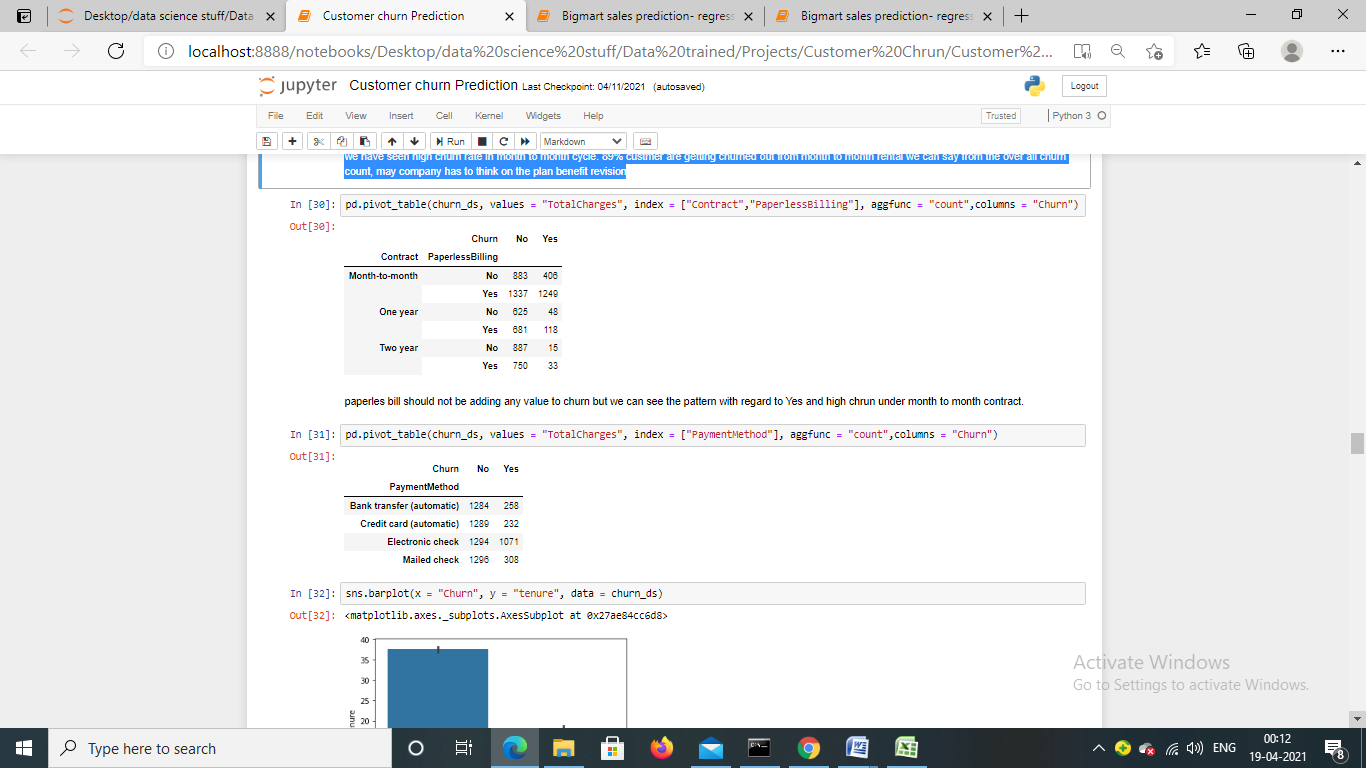
Charges are high of Fiber optic connection as compared to DSL (digital subscriber line), people who no phone service are using DSL connection for internet.

Fiber optic connection is generating higher revenue for company.



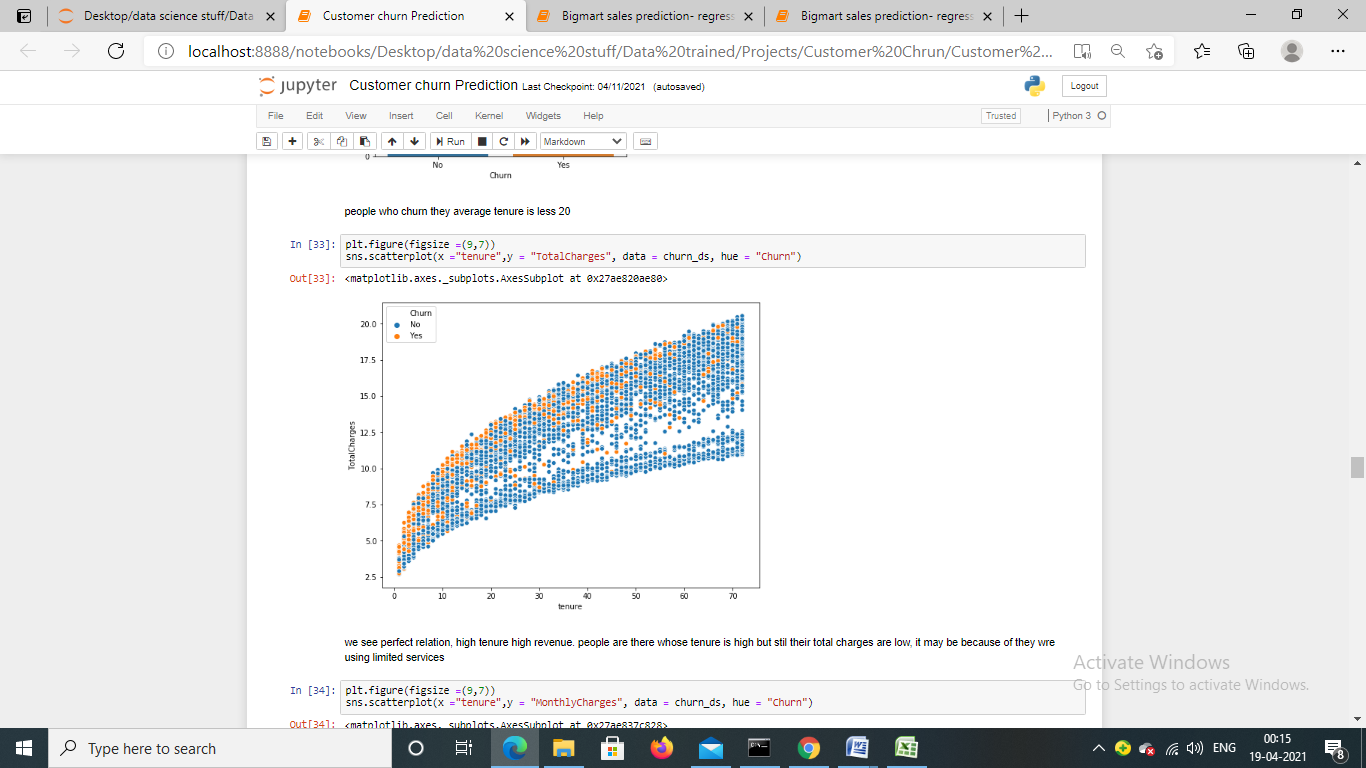
We have seen high churn rate in month to month bill cycle. 89% customer are getting churned out from month to month rental, we can say from the overall churn count, may company has to think on the plan benefit revision

**Payment method wise analysis:-**

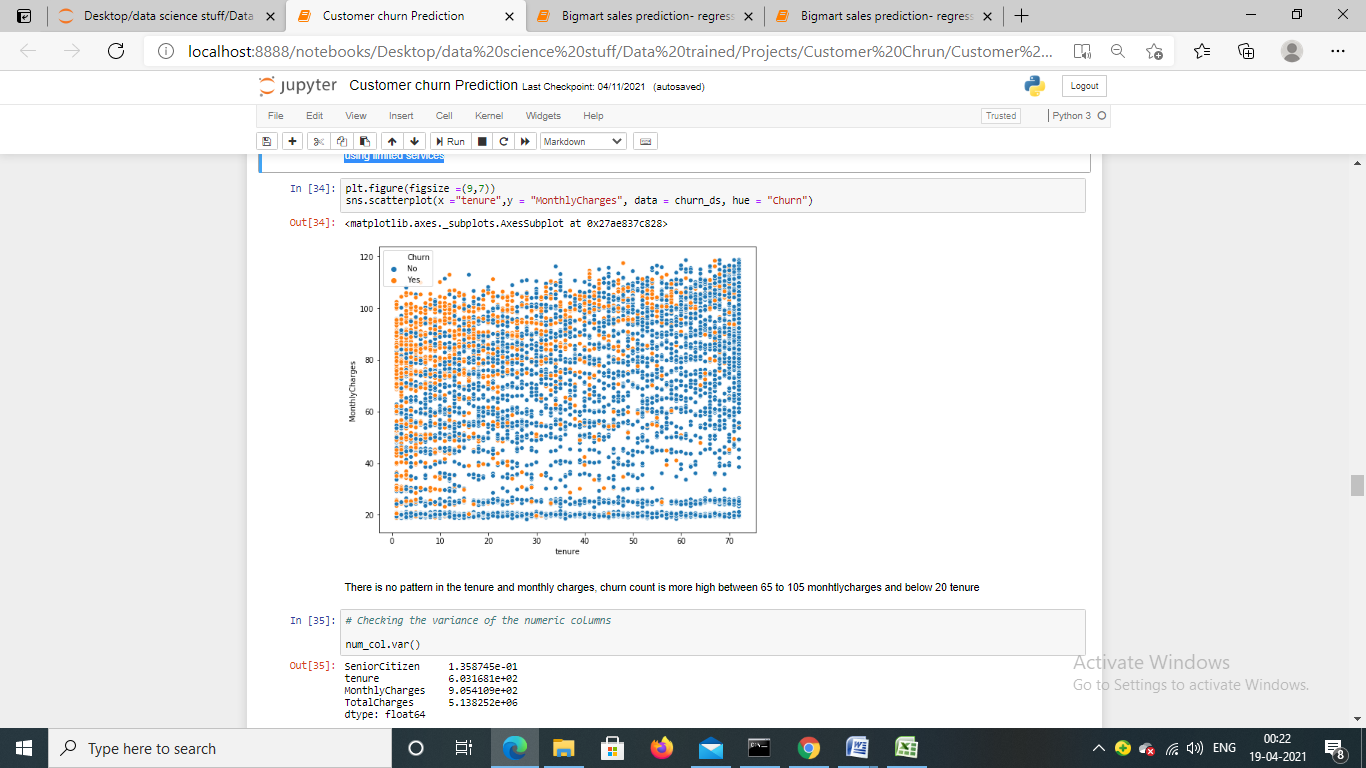


Maximum churn rate is observed from who were making payment through electronic check. Other methods were also popular but we see high churn rate in electronic check. Assuming that people who wanted to use service for less time or they wanted to see how service is being provided probably they would use electronic method for monthly payment rather than going for auto debit.

Let’s check the relationship between tenure and Total charges:



We see perfect relation, high tenure high revenue. We have also seen that people whose tenure is high but still their total charges are low, it may be because of they were using limited services but high tenure are generating high revenue.



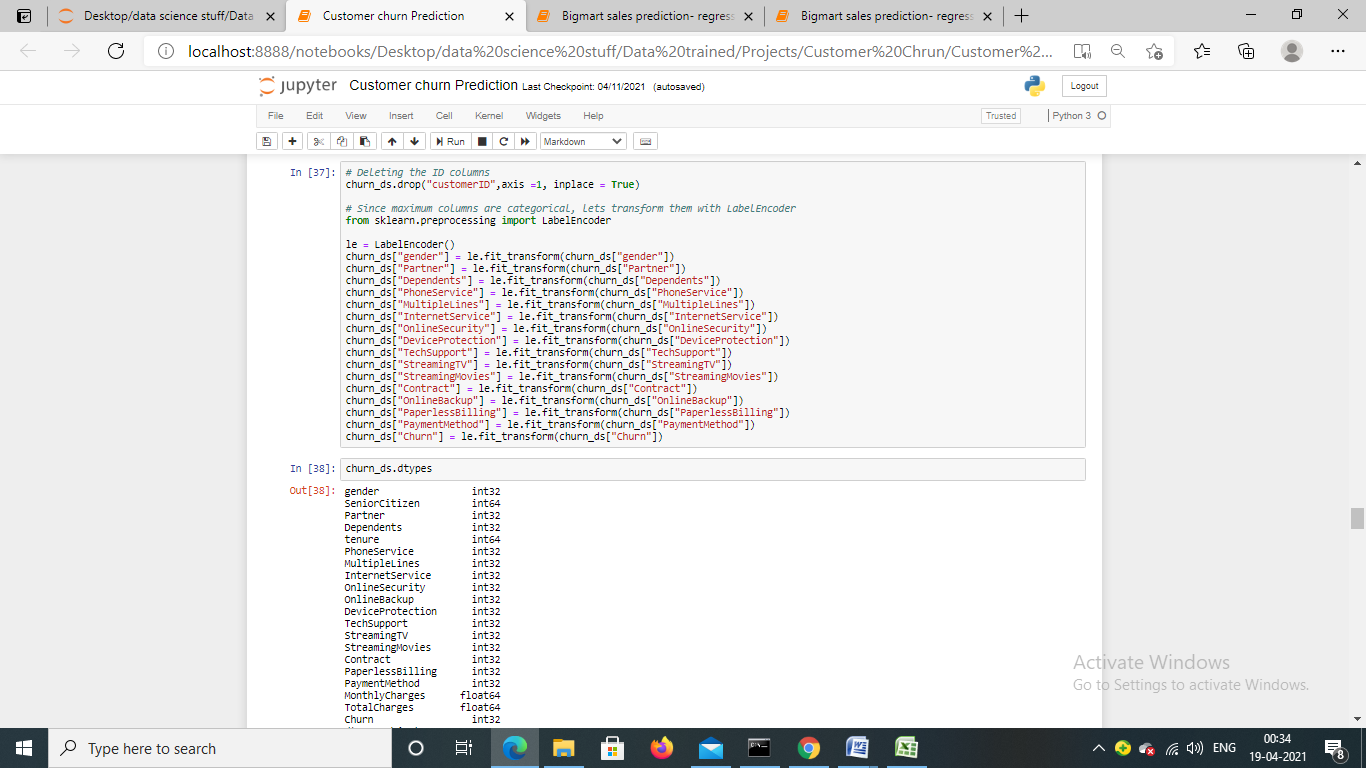
I have also tried checking the relation between monthly charges and tenure but i didn’t find any relation.

* **Pre-Processing Pipeline**

**Label Encoding**

We have done some analysis and found interesting observations, missing value and outlier have been checked and treated accoringly.

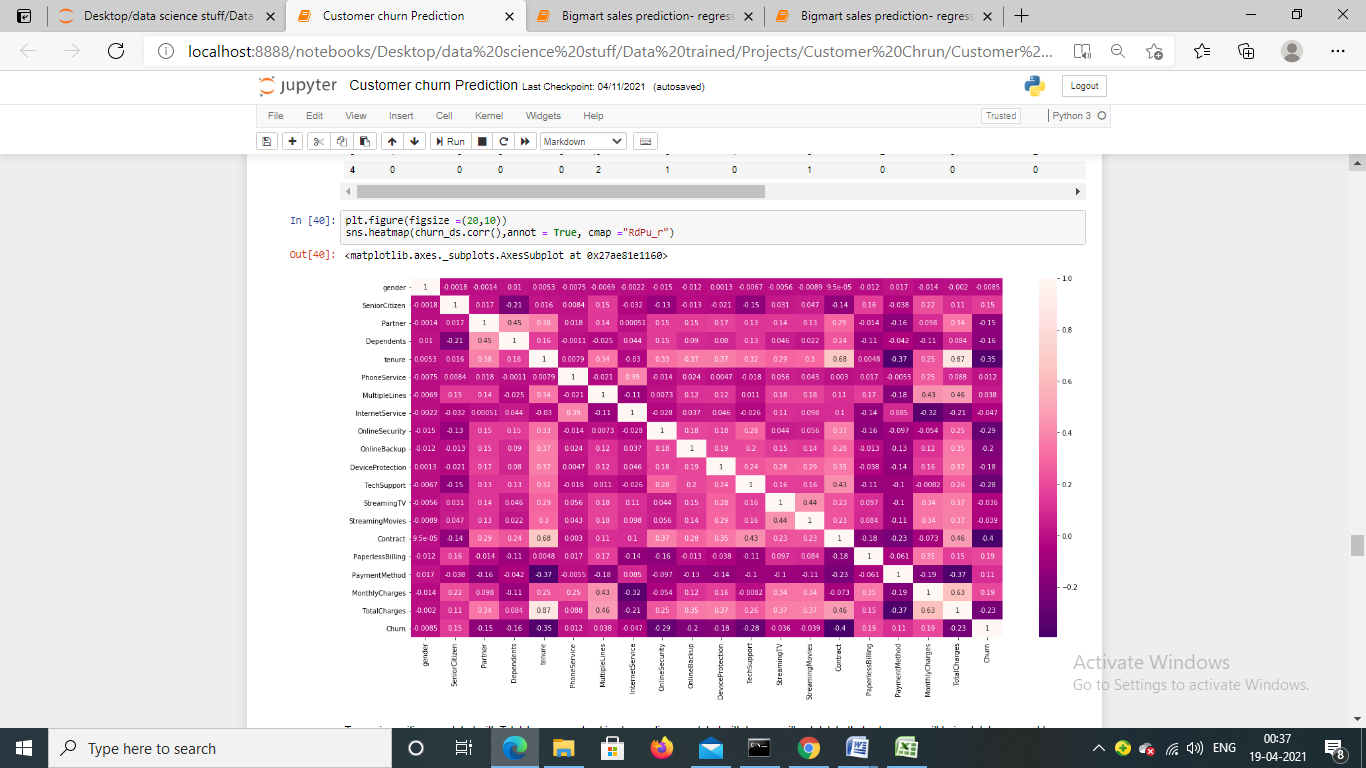
Label encoding, machine learning algorithm takes numerical input for the learning, thus it becomes important that data should be converted into number. We have a few categorical columns let’s convert them.



I have deleted Customer Id column as it is not useful for model building

Here, i am using sklearn’s LabelEncoder method to encode categorical features, we have 18 columns like that including target variable and all are processed.

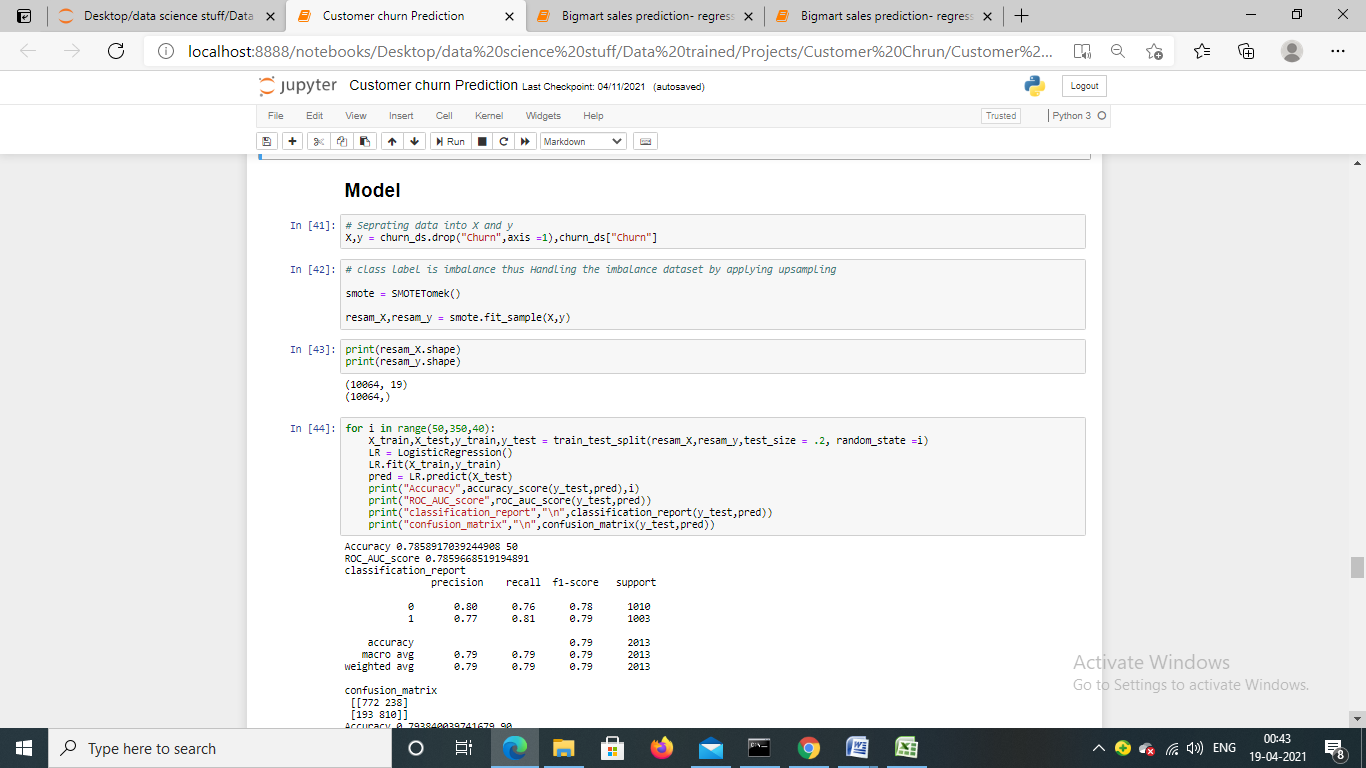
**Let’s check the correlation plot:-**

****

We have found that tenure column is positive correlated with Total charges, contract columns is also positive correlated with tenure. Let’s move further for model building

* **Building Machine Learning Models**

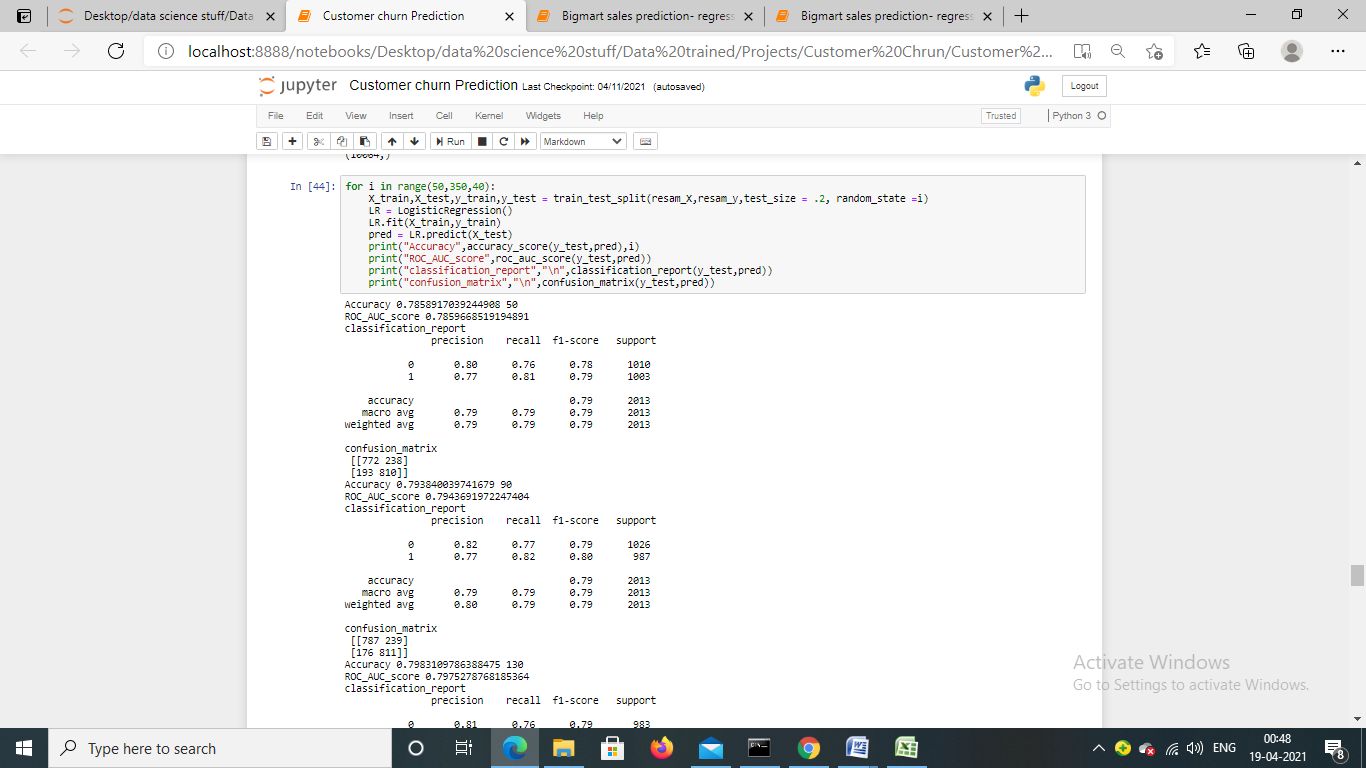
We saw in the above section while checking the proposition of target features, it wasn’t equal, it was data imbalance problem and we discussed that would see the solution so here is the solution:-



I have separated data into X and Y and after I have tried handling class imbalance problem by using SMOTE method with up-sampling technique, now we have equal class label.

**Applying ML algorithm**

We will use multiple machine learning algorithms to train model and pick the best one.



I am using Logistic regression as base model and running for loop for finding the best random state, let’s check the result of the model:-

Accuracy 0.7858917039244908 50

ROC\_AUC\_score 0.7859668519194891

classification\_report

precision recall f1-score support

0 0.80 0.76 0.78 1010

1 0.77 0.81 0.79 1003

accuracy 0.79 2013

macro avg 0.79 0.79 0.79 2013

weighted avg 0.79 0.79 0.79 2013

confusion\_matrix

[[772 238]

[193 810]]

Accuracy 0.793840039741679 90

ROC\_AUC\_score 0.7943691972247404

classification\_report

precision recall f1-score support

0 0.82 0.77 0.79 1026

1 0.77 0.82 0.80 987

accuracy 0.79 2013

macro avg 0.79 0.79 0.79 2013

weighted avg 0.80 0.79 0.79 2013

confusion\_matrix

[[787 239]

[176 811]]

Accuracy 0.7983109786388475 130

ROC\_AUC\_score 0.7975278768185364

classification\_report

precision recall f1-score support

0 0.81 0.76 0.79 983

1 0.79 0.83 0.81 1030

accuracy 0.80 2013

macro avg 0.80 0.80 0.80 2013

weighted avg 0.80 0.80 0.80 2013

confusion\_matrix

[[751 232]

[174 856]]

Accuracy 0.8022851465474417 170

ROC\_AUC\_score 0.8021825612272093

classification\_report

precision recall f1-score support

0 0.82 0.77 0.80 1003

1 0.79 0.83 0.81 1010

accuracy 0.80 2013

macro avg 0.80 0.80 0.80 2013

weighted avg 0.80 0.80 0.80 2013

confusion\_matrix

[[775 228]

[170 840]]

Accuracy 0.7824143070044709 210

ROC\_AUC\_score 0.7824641672589433

classification\_report

precision recall f1-score support

0 0.80 0.75 0.78 1008

1 0.76 0.82 0.79 1005

accuracy 0.78 2013

macro avg 0.78 0.78 0.78 2013

weighted avg 0.78 0.78 0.78 2013

confusion\_matrix

[[755 253]

[185 820]]

Accuracy 0.7878787878787878 250

ROC\_AUC\_score 0.7882260959184036

classification\_report

precision recall f1-score support

0 0.82 0.74 0.78 1014

1 0.76 0.83 0.80 999

accuracy 0.79 2013

macro avg 0.79 0.79 0.79 2013

weighted avg 0.79 0.79 0.79 2013

confusion\_matrix

[[752 262]

[165 834]]

Accuracy 0.7968206656731247 290

ROC\_AUC\_score 0.7973107186225824

classification\_report

precision recall f1-score support

0 0.83 0.75 0.79 1018

1 0.77 0.84 0.80 995

accuracy 0.80 2013

macro avg 0.80 0.80 0.80 2013

weighted avg 0.80 0.80 0.80 2013

confusion\_matrix

[[768 250]

[159 836]]

Accuracy 0.8072528564331843 330

ROC\_AUC\_score 0.8071201810029792

classification\_report

precision recall f1-score support

0 0.83 0.78 0.80 1002

1 0.79 0.84 0.81 1011

accuracy 0.81 2013

macro avg 0.81 0.81 0.81 2013

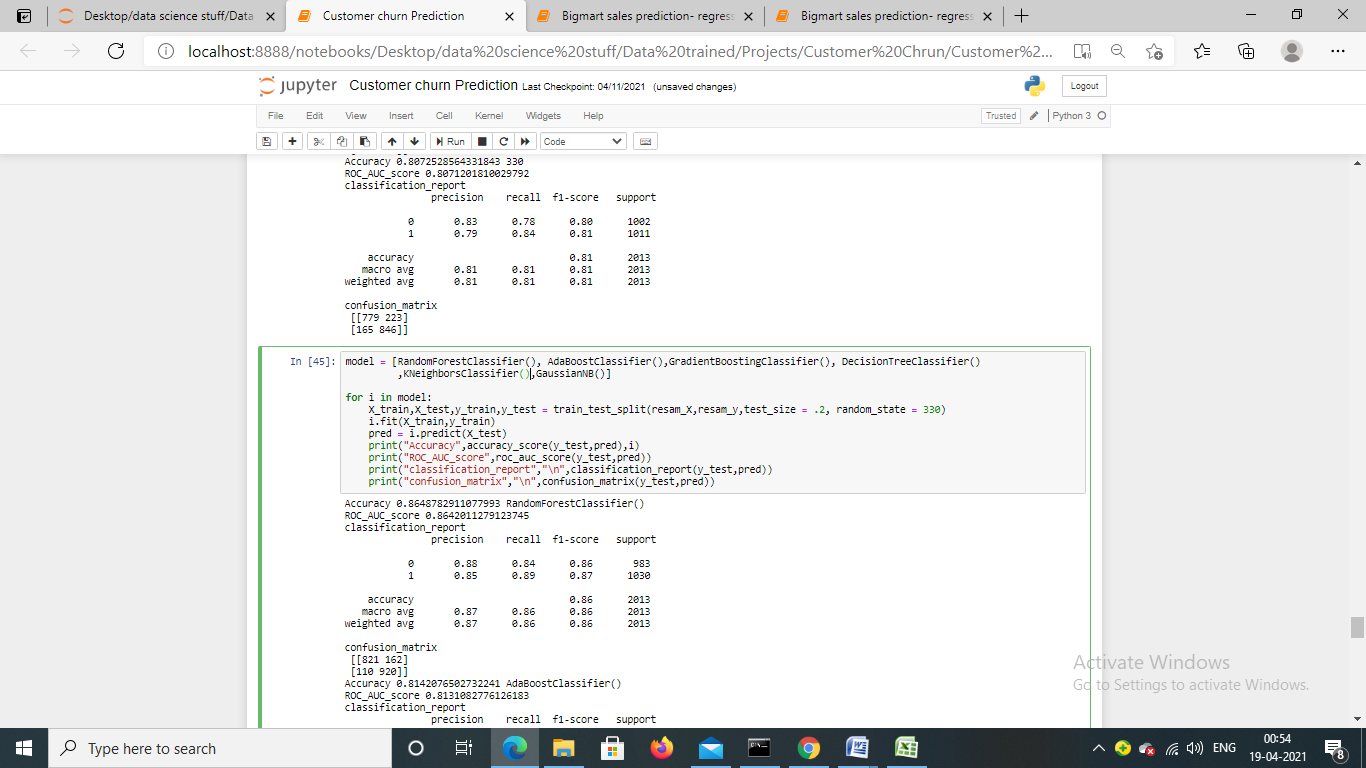
weighted avg 0.81 0.81 0.81 2013

confusion\_matrix

[[779 223]

[165 846]

Random state 330 is giving highest ROC\_AUC\_score and f1 score is also better than other so we will select random state 330 and trained other models.



We are using 6 machine learning algorithms, we will check that which one is working well on the dataset and will pick that one for model saves and hyper parameter tuning. Let’s check the result:-

Accuracy 0.8648782911077993 RandomForestClassifier()

ROC\_AUC\_score 0.8642011279123745

classification\_report

precision recall f1-score support

0 0.88 0.84 0.86 983

1 0.85 0.89 0.87 1030

accuracy 0.86 2013

macro avg 0.87 0.86 0.86 2013

weighted avg 0.87 0.86 0.86 2013

confusion\_matrix

[[821 162]

[110 920]]

Accuracy 0.8142076502732241 AdaBoostClassifier()

ROC\_AUC\_score 0.8131082776126183

classification\_report

precision recall f1-score support

0 0.84 0.77 0.80 983

1 0.79 0.86 0.83 1030

accuracy 0.81 2013

macro avg 0.82 0.81 0.81 2013

weighted avg 0.82 0.81 0.81 2013

confusion\_matrix

[[753 230]

[144 886]]

Accuracy 0.8286140089418778 GradientBoostingClassifier()

ROC\_AUC\_score 0.827719779948444

classification\_report

precision recall f1-score support

0 0.85 0.79 0.82 983

1 0.81 0.87 0.84 1030

accuracy 0.83 2013

macro avg 0.83 0.83 0.83 2013

weighted avg 0.83 0.83 0.83 2013

confusion\_matrix

[[776 207]

[138 892]]

Accuracy 0.8201689021361153 DecisionTreeClassifier()

ROC\_AUC\_score 0.8197922942448815

classification\_report

precision recall f1-score support

0 0.82 0.80 0.81 983

1 0.82 0.84 0.83 1030

accuracy 0.82 2013

macro avg 0.82 0.82 0.82 2013

weighted avg 0.82 0.82 0.82 2013

confusion matrix

[[790 193]

[169 861]]

Accuracy 0.8142076502732241 KNeighborsClassifier()

ROC\_AUC\_score 0.8118085116890044

classification\_report

Precision recall f1-score support

0 0.89 0.71 0.79 983

1 0.77 0.91 0.83 1030

Accuracy 0.81 2013

Macro avg 0.83 0.81 0.81 2013

Weighted avg 0.83 0.81 0.81 2013

confusion matrix

[[697 286]

[ 88 942]]

Accuracy 0.7848981619473423 GaussianNB()

ROC\_AUC\_score 0.7833766259419848

classification report

Precision recall f1-score support

0 0.82 0.72 0.77 983

1 0.76 0.85 0.80 1030

Accuracy 0.78 2013

Macro avg 0.79 0.78 0.78 2013

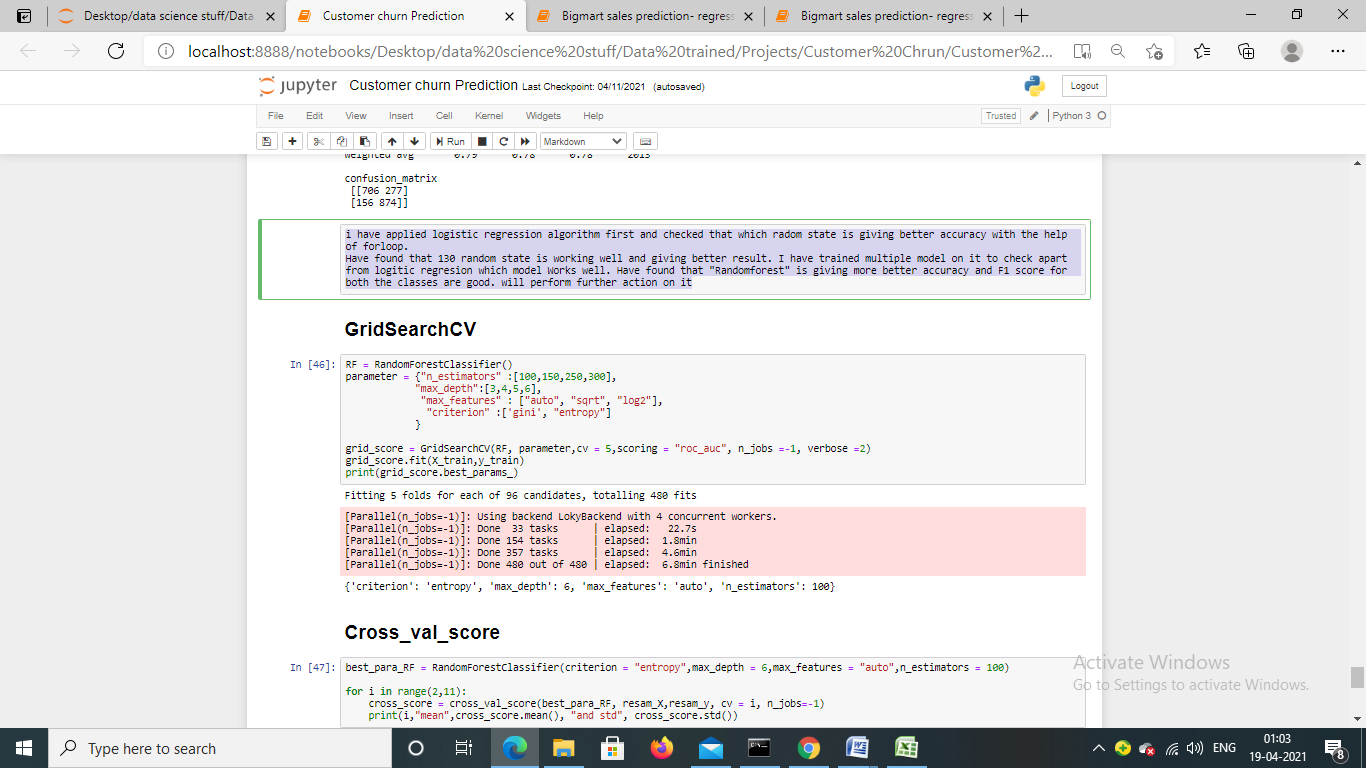
Weighted avg 0.79 0.78 0.78 2013

confusion matrix

[[706 277]

[156 874]]

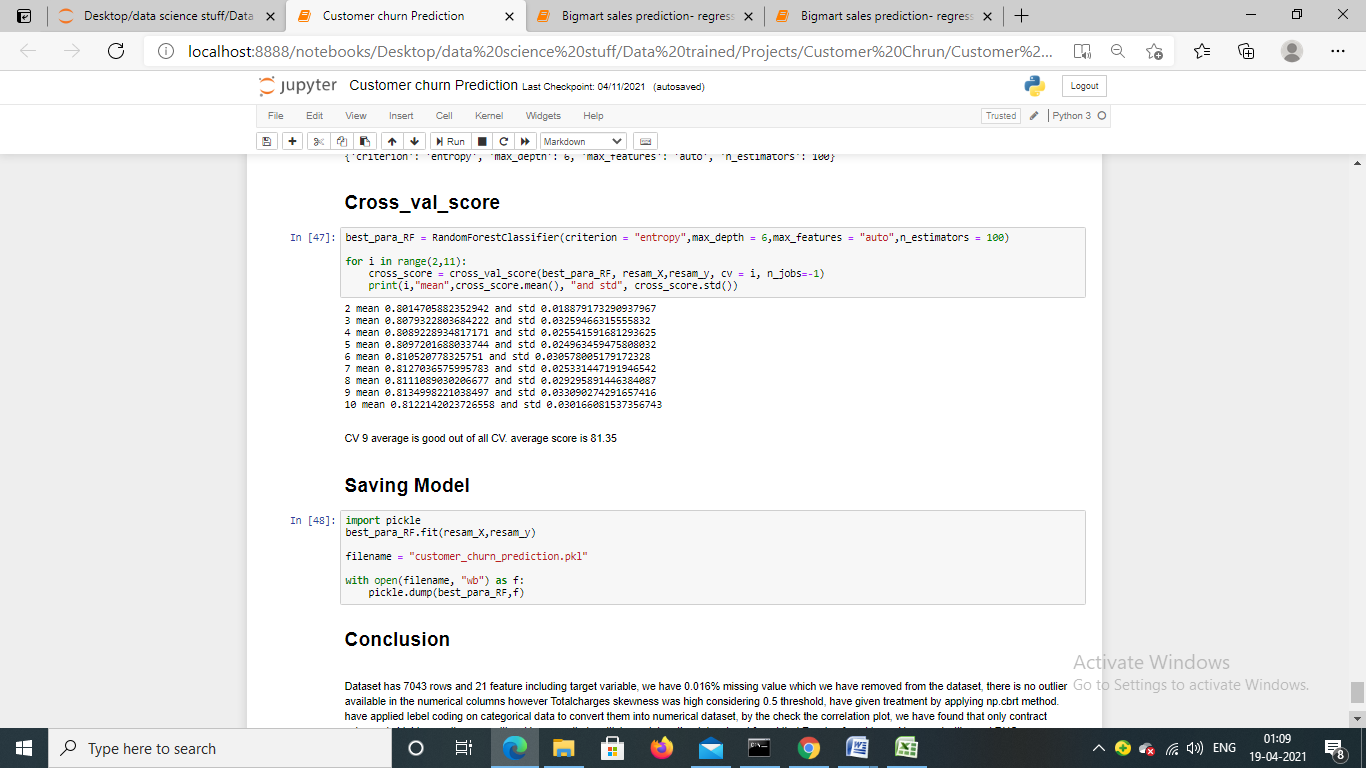
As we have checked, random forest is working better than other ML models, its ROC\_AUC\_score is 0.86 which means that our model working good and explain 86% of the data accurately . We are picking this one as final and moving further, we will perform Grid search CV and cross validation with random forest.



I have used Grid search CV to find best parameter of random forest.

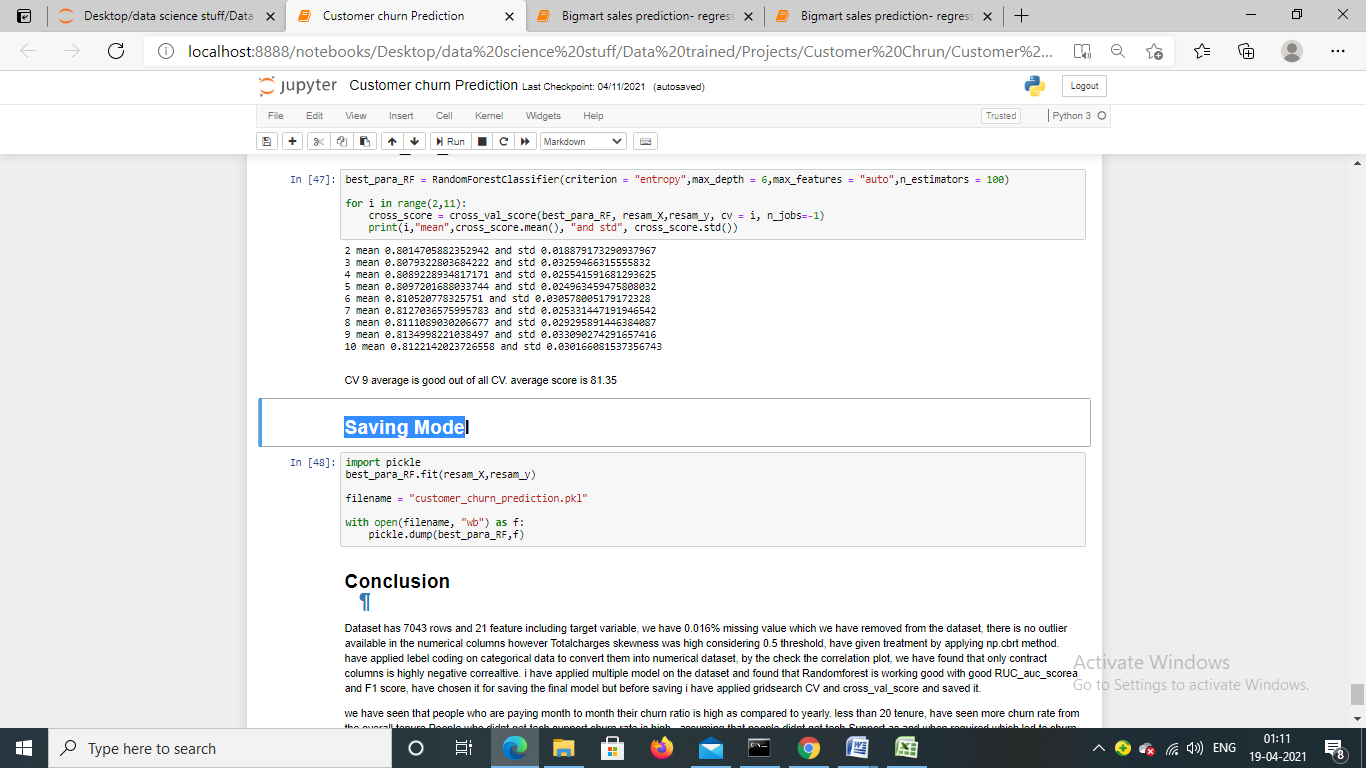
Now, we are going to perform cross validation, cross validation trains model on entire data point so that model learns every pattern we also take average score of all the models along with standard deviation to check how well model is working . We get score by using mean () function and we check the standard deviation as well of the score by std (). Here is the code snippet:-

**Cross validation:-**

****

We can see that cross validation 9th score is greater than CV, now we are saving model.

**Model saving**



We are saving model by using python’s pickle library. It will be used further for the prediction.

**Conclusion:-**

Dataset has 7043 rows and 21 feature including target variable, we have 0.016% missing value which we have removed from the dataset, there is no outlier available in the numerical columns however Total charges skewness was high considering 0.5 threshold, have given treatment by applying np.cbrt method. I have applied label coding on categorical data to convert them into numerical dataset, from correlation plot; we have found that an only contract column is highly negative correlative. i have applied multiple model on the dataset and found that Random forest is working good with good RUC\_auc\_score and F1 score, have chosen it for saving the final model but before saving i have applied Grid search CV and cross\_val\_score and saved it.

We have seen that people who are paying month to month their churn ratio is high as compared to yearly. Tenure less than 20, have seen more churn rate from the overall tenure. People who didn’t get tech support churn rate is high for that and assuming that people didn’t get tech support as and when required which led to churn.